

# Détection et étiquetage automatique d'image

Par

Joël Brou Boni

P-SAT

Mr Elod Egyed-Zsigmond



## COMMENT UTILISER LE DETECTEUR AUTOMATIQUE D'IMAGE ?

Il s'agit tout d'abord d'avoir toutes les librairies indispensables à son utilisation.

J'ai exécuté ce programme sur **Mac et Linux** sans problème, il devrait en être pour **Windows**.

Si cela ne marche pas sur cet OS, il s'agira certainement de dépendances liées à CUDA et Tensorflow.

**Python 3.5–3.8** est directement compatible avec TensorFlow.

La compatibilité avec Python 3.8 nécessite TensorFlow 2.2 ou version ultérieure.

J'ai pour cela créé un fichier requirements.txt présent dans le répertoire **Image detection** récapitulant toutes les dépendances requises.

Il faudra ensuite exécuter la ligne suivante dans le répertoire **Image detection**.

**pip install -r requirements.txt**

## UN RESEAU DE NEURONES EXPLOITANT LA YOLOV 3

- Le fichier de poids **YOLOV3** permettant l'étiquetage des objets étant trop lourd pour être transmis en zip, il est possible de les télécharger à cette adresse <https://pjreddie.com/media/files/yolov3.weights>.
- Une fois ce fichier **yolov3.weights** téléchargé et placé dans le répertoire courant, vous devez avoir à présent dans le répertoire **Image detection** les fichiers **yolov3.weights**, **detection.py** et **requirements.txt**

Enfin, il y'a à disposition des images(.png) afin de tester l'efficacité de notre programme.

## FONCTIONNEMENT DU PROGRAMME

Une fois dans le dossier **Image detection**, rien de plus simple pour étiqueter notre image que d'exécuter une commande de la forme suivante :

**python detection.py image.jpg**

avec **image.jpg** représentant le nom et extension de l'image à analyser

Cette image doit se trouver dans le répertoire **Image detection**

Après 15 secondes en moyenne pour la prédiction de notre réseau à partir d'une image et 1 à 2 minutes au lancement de notre projet pour charger notre modèle,

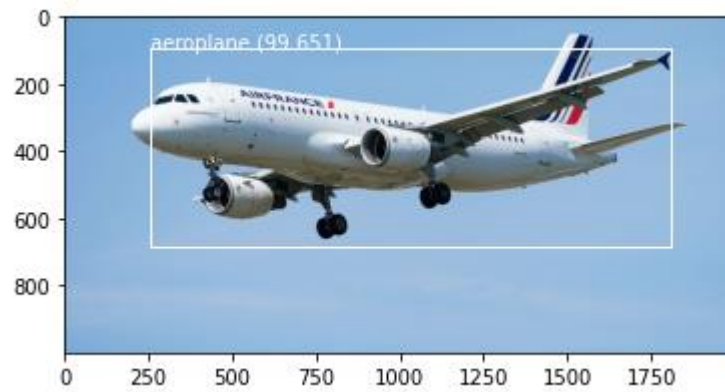
Vous aurez dans la console des tags concernant des objets identifiés sur l'image et le pourcentage de fiabilité calculé grâce au réseau de neurones YOLOV3 que cet objet soit présent.

Et vous aurez dans le dossier **Image detection** la création d'un fichier nommé "annotated\_image.jpg" qui encadrera les objets identifiés avec les tags associés.

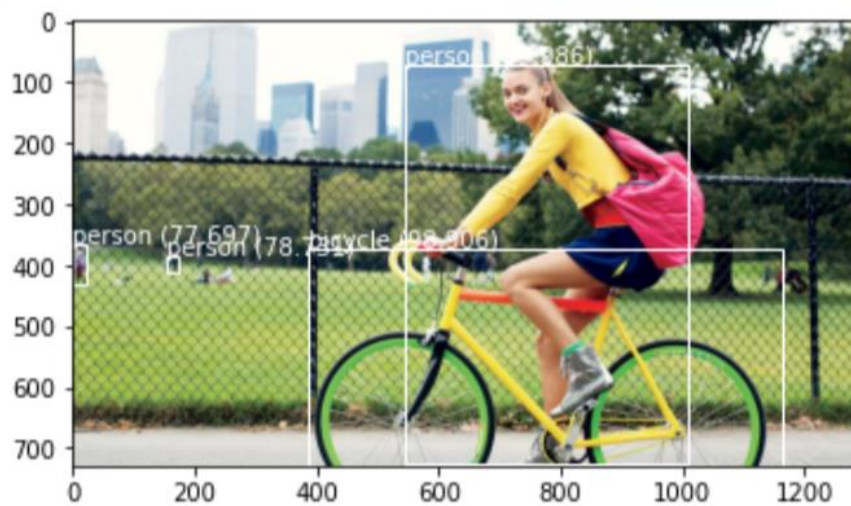
Exemple : **annotated\_avion.jpg**

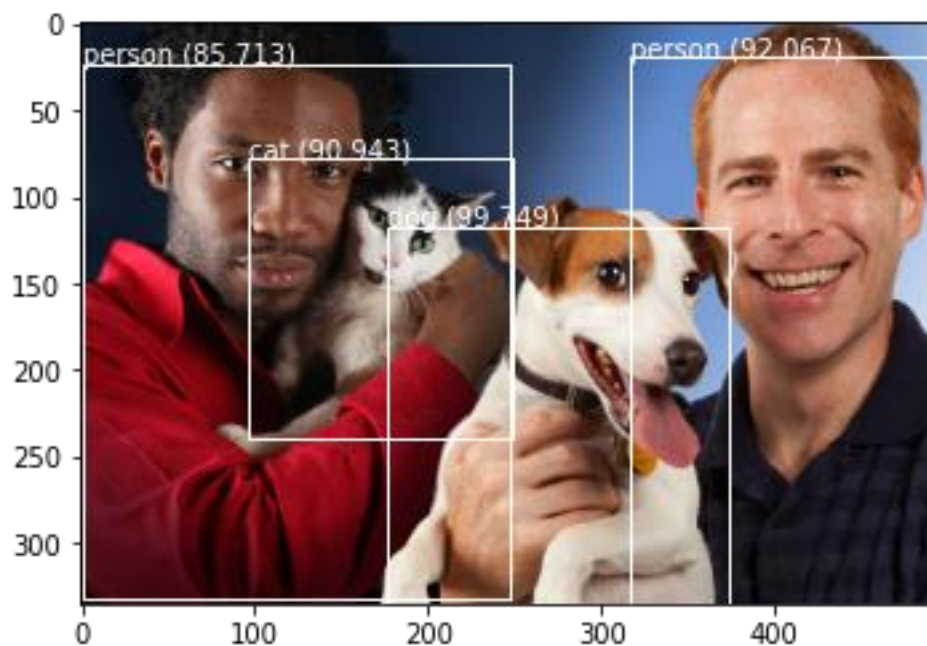
**annotated\_personbike.jpg, annotated\_greta.jpg**

aeroplane 99.65118169784546



person 99.98559951782227  
bicycle 98.90583753585815  
person 77.69652009010315  
person 78.73073816299438





Pour supprimer les images annotées, rien de plus simple que d'exécuter la ligne suivante.

**rm -rf annotated\***

Voici l'ensemble des tags couverts par **Yolov3** et sur lesquels le programme se montre performant

```
labels = ["person", "bicycle", "car", "motorbike", "aeroplane", "bus", "train",
"truck", "boat", "traffic light", "fire hydrant", "stop sign", "parking meter",
"bench", "bird", "cat", "dog", "horse", "sheep", "cow", "elephant", "bear", "zebra",
"giraffe", "backpack", "umbrella", "handbag", "tie", "suitcase", "frisbee", "skis",
"snowboard", "sports ball", "kite", "baseball bat", "baseball glove", "skateboard",
"surfboard", "tennis racket", "bottle", "wine glass", "cup", "fork", "knife",
"spoon", "bowl", "banana", "apple", "sandwich", "orange", "broccoli", "carrot",
"hot dog", "pizza", "donut", "cake", "chair", "sofa", "pottedplant", "bed",
"diningtable", "toilet", "tvmonitor", "laptop", "mouse", "remote", "keyboard",
"cell phone", "microwave", "oven", "toaster", "sink", "refrigerator", "book",
"clock", "vase", "scissors", "teddy bear", "hair drier", "toothbrush"]
```

Pour gagner du temps, j'ai aussi écrit un script permettant de télécharger directement un nombre d'images spécifié d'un ou plusieurs thèmes directement pour automatiser le téléchargement d'images pour les tests que vous pourrez utiliser si vous le souhaitez, par exemple dans Jupyter.

**pip install git+<https://github.com/JoecClinton1/google-images-download.git>**(dans le terminal)

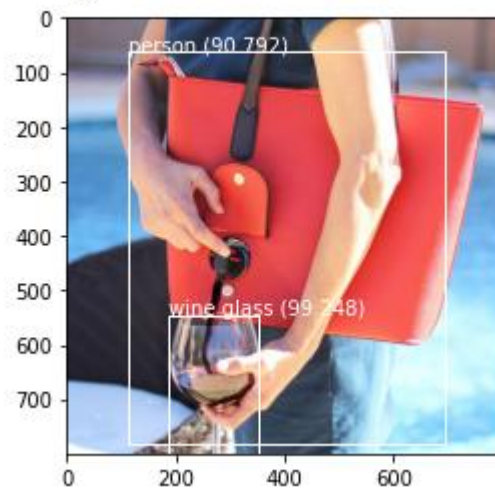
Dans Jupyter:

```
from google_images_download import google_images_download
response = google_images_download.googleimagesdownload()
arguments = {"keywords":"Banana, Orange","limit":10, "format":"jpg",
"print_urls":False}
image = response.download(arguments)
```

Vous aurez ainsi 5 images de Bananes et 5 d'oranges dans des dossiers ayant les mêmes noms dans vos téléchargements.

Vous pourrez ensuite transférer les images dans le dossier **Image detection** et ensuite les soumettre au programme.

```
person 90.79150557518005
wine glass 99.2480456829071
```



**Merci de votre attention !**